

NetCDF Utility Library: User's Documentation

JULES KOUATCHOU AND MAHARAJ BHAT

NASA Goddard Space Flight Center

Software Integration and Visualization Office

Advanced Software Technology Group

Code 610.3

Greenbelt, MD 20771

Jules.Kouatchou.1@gsfc.nasa.gov & Maharaj.K.Bhat.1@gsfc.nasa.gov

June 3, 2005

Abstract

We introduce a netCDF utility library to manipulate netCDF files. Routines in the library were written to simplify netCDF calls and to hide from the user error checking procedures. The library has successfully been implemented in several applications (GMI for instance) and was ported into different platforms (HP SC45, SGI Altix, workstations). Practical examples and the list of available routines are provided to show how the library could be employed.

Contents

1	Introduction	4
2	How to Use the Library	4
2.1	Structure	4
2.2	Compiling Source Files	5
2.3	Creating NetCDF Files	5
2.4	Reading NetCDF Files	8
A	Routine/Function Prologues	11
A.1	Fortran: Module Interface m_Do_Err_Out (Source File: m_do_err_out.f90)	11
A.1.1	Do_Err_Out	11
A.2	Fortran: Module Interface m_ncdf_io_checks (Source File: m_ncdf_io_checks.f90)	12
A.2.1	Ncdoes_Udim_Exist (Source File: m_ncdf_io_checks.f90)	12
A.2.2	Ncdoes_Var_Exist (Source File: m_ncdf_io_checks.f90)	13
A.3	Fortran: Module Interface m_ncdf_io_close (Source File: m_ncdf_io_close.f90)	14
A.3.1	Nccl	14
A.3.2	Nccl_Noerr	15
A.4	Fortran: Module Interface m_ncdf_io_create (Source File: m_ncdf_io_create.f90)	15
A.4.1	Nccr_Wr	16
A.4.2	Ncdo_Sync	16
A.5	Fortran: Module Interface m_ncdf_io_define (Source File: m_ncdf_io_define.f90)	17
A.5.1	NcDef_dimension	17
A.5.2	NcDef_variable	18
A.5.3	NcDef_var_attributes	19
A.5.4	NcDef_glob_attributes	20
A.5.5	NcSetFill	20
A.5.6	NcEnd_def	21
A.6	Fortran: Module Interface m_ncdf_io_get_dimlen (Source File: m_ncdf_io_get_dimlen.f90)	21
A.6.1	Ncget_Dimlen	22
A.6.2	Ncget_Unlim_Dimlen	22
A.7	Fortran: Module Interface m_ncdf_io_handle_err (Source File: m_ncdf_io_handle_err.f90)	23
A.7.1	Nchandle_Err	23
A.8	Fortran: Module Interface m_ncdf_io_open (Source File: m_ncdf_io_open.f90)	24
A.8.1	Ncop_Rd	24
A.8.2	Ncop_Wr	25
A.9	Fortran: Module Interface m_ncdf_io_read (Source File: m_ncdf_io_read.f90)	26
A.9.1	Ncrd_Scal	26
A.9.2	Ncrd_Scal_Int	27
A.9.3	Ncrd_1d	28
A.9.4	Ncrd_1d_Int	28
A.9.5	Ncrd_2d	29
A.9.6	Ncrd_2d_Int	30
A.9.7	Ncrd_3d	31
A.9.8	Ncrd_3d_Int	31
A.9.9	Ncrd_4d	32
A.9.10	Ncrd_5d	33

A.9.11 Ncrd_1d_Char	34
A.9.12 Ncrd_2d_Char	35
A.10 Fortran: Module Interface m_ncdf_io_write (Source File: m_ncdf_io_write.f90)	36
A.10.1 Ncwr_Scal	36
A.10.2 Ncwr_Scal_Int	37
A.10.3 Ncwr_1d	38
A.10.4 Ncwr_1d_Int	38
A.10.5 Ncwr_2d	39
A.10.6 Ncwr_2d_Int	40
A.10.7 Ncwr_3d	41
A.10.8 Ncwr_3d_Int	41
A.10.9 Ncwr_4d	42
A.10.10 Ncwr_5d	43
A.10.11 Ncwr_6d	44
A.10.12 Ncwr_1d_Char	44
A.10.13 Ncwr_2d_Char	45

1 Introduction

This document describes a netCDF[2] utility library written to perform operations on netCDF files. These operations include for instance

- Creating a netCDF file
- Defining dimensions
- Defining a variable
- Defining attribute values for variables
- Writing values of variables
- Reading variables from a file.

The main advantage of using the library is that it simplifies netCDF calls by doing checking procedures in the background and by reducing the argument list of some netCDF routines. An error message is printed if something breaks: variable not defined, variable does not exist in a file, improper dimension of a variable, etc.

Some of the utility routines contained here were taken out of the GMI code [1] and were initially written by John Tannahill (LLNL, jrt@llnl.gov). We expanded on John Tannahill's work by adding new utility routines and including all routines in Fortran 90 modules.

This library has been implemented in the GMI code and other codes. It has been installed and tested on *halem*, *palm*, and a linux based computer. We provide below information on how to use it and also the list of routines (and their arguments) available in the library.

2 How to Use the Library

2.1 Structure

The netCDF utility library can be obtained by anonymous ftp from dirac.gsfc.nasa.gov:

```
pub/gmidata/Utils/netCDF_utils.tar
```

To untar "netCDF_utils.tar", type

```
tar xvf netCDF_utils.tar
```

and you will obtain the directory netCDF_utils/ that has the following structure:

README	README file
doc/	documentation directory
example/	directory providing two sample codes
incl_mods/	directory where .mod files reside after compilation.
lib/	directory for the library
src/	source files (Fortran 90)

2.2 Compiling Source Files

To generate the library

- Go to the src/ directory
 - Edit the Makefile file and set the variables "INCLUDES_NETCDF" and "NETCDF_LIBS" to provide (on your system) the paths for the netCDF include files and netCDF library respectively.
 - Type the command "make".
 - .mod files will appear in incl_mods/ and the utility library file in lib/.

In the next two subsections, we provide two sample Fortran programs implementing the use of the library. the two programs are available in the example/ directory and can be compiled and run (a README and Makefile files are provided).

2.3 Creating NetCDF Files

The main steps for creating a new netCDF file and writing data in it are:

As an example, assume that you want to write a Fortran program that computes the surface pressure (2D variable) and the temperature (3D variable). You want to write these data in a netCDF file.

```
!!!!!! ! This program computes the surface pressure (psf, 2D variable) and the
! the temperature (kel, 3D variable) and write them out in a netCDF
! file (my_filename.nc). It is assumed that the user provides the
! routines to compute psf and kel as well as how to determine the
! pressure levels.
!!!!!!
```

```

!
! Modules from the utility library
use m_ncdf_io_create
use m_ncdf_io_define
use m_ncdf_io_write
use m_ncdf_io_close
!

implicit none
include "netcdf.inc"
!

integer, parameter :: ilong = 72 ! number of longitude grid points
integer, parameter :: ilat = 46 ! number of latitude grid points
integer, parameter :: ivert = 46 ! number of pressure levels
!

integer          :: lon_id(1), lat_id(1), prs_id(1)
integer          :: cnt1d(1), cnt2d(2), cnt3d(3)
integer          :: strt1d(1), strt2d(2), strt3d(3)
integer          :: longdeg, latdeg
real*8           :: longdat(ilong), latdat(ilat)
real*8           :: prsdat(ivert)
real*8           :: psf (ilong, ilat)      ! surface pressure
real*8           :: kel (ilong, ilat, ivert) ! temperature
!

integer          :: ncid, varid
integer          :: var2(2), var3(3)
integer          :: omode, i
!

! Create the netCDF file
call Nccr_Wr (ncid, 'my_filename.nc')
!

! Define the dimensions
call NcDef_dimension(ncid,'lon_dim',ilong,lon_id(1))
call NcDef_dimension(ncid,'lat_dim',ilat ,lat_id(1))
call NcDef_dimension(ncid,'prs_dim',ivert,prs_id(1))
!

! Define the variables and variable attributes
!

call NcDef_variable(ncid,'lon_dim',NF_FLOAT,1,lon_id,varid)
call NcDef_var_attributes(ncid,varid,'long_name','Longitude')
call NcDef_var_attributes(ncid,varid,'units','degree_east')
!

call NcDef_variable(ncid,'lat_dim',NF_FLOAT,1,lat_id,varid)
call NcDef_var_attributes(ncid,varid,'long_name','Latitude')
call NcDef_var_attributes(ncid,varid,'units','degree_north')
!

call NcDef_variable(ncid,'prs_dim',NF_FLOAT,1,prs_id,varid)
call NcDef_var_attributes(ncid,varid,'long_name','Pressure')
call NcDef_var_attributes(ncid,varid,'units','millibars')
!
```

```

!
var2(:) = (/lon_id(1), lat_id (1) /)
call NcDef_variable(ncid,'psf',NF_FLOAT,2,var2,varid)
call NcDef_var_attributes (ncid, varid, 'long_name', 'Surface Pressure')
call NcDef_var_attributes (ncid, varid, 'units', 'millibars')

!
var3(:) = (/lon_id(1), lat_id (1), prs_id(1) /)
call NcDef_variable(ncid,'kel',NF_FLOAT,3,var3,varid)
call NcDef_var_attributes (ncid, varid, 'long_name', 'Temperature')
call NcDef_var_attributes (ncid, varid, 'units', 'degK')

!
! Define global attributes
!

call NcDef_glob_attributes(ncid,'title','Surf. Pres. & Temp. output')
call NcDef_glob_attributes(ncid,'history','Initial file for testing - 051005')
call NcDef_glob_attributes(ncid,'Conventions','COARDS')
call NcSetFill(ncid, NF_NOFILL, omode)

!
! Closing the definition section
call NcEnd_def(ncid)

!
! Writing longitude data point
longdeg = 360/ilong
if (mod(360, ilong) .ne. 0) longdeg = longdeg + 1
do i = 1, ilong
    longdat(i) = i*longdeg
enddo

!
strt1d(1) = 1
cnt1d (1) = ilong
call Ncwr_1d (longdat, ncid, 'lon_dim', strt1d, cnt1d)

!
! Writing latitude data point
latdeg = 180/ilat
if (mod(180, ilat) .ne. 0) latdeg = latdeg + 1
do i = 1, ilong
    latdat(i) = -90 + (i-0.5)*latdeg
enddo

!
strt1d(1) = 1
cnt1d (1) = ilat
call Ncwr_1d (latdat, ncid, 'lat_dim', strt1d, cnt1d)

!
! Writing pressure levels
! enter your method to compute pressure levels

do i = 1, ivert
    prsdat(i) = 1000.00 - (i-1)*(920.00/ivert)

```

```

    end do
!
strt1d(1) = 1
cnt1d (1) = invert
call Ncwr_1d (prsdat, ncid, 'prs_dim', strt1d, cnt1d)
!
! Include here you code to compute
!     the surface pressure 'psf', and
!     the temperature 'kel'.
!
! Write the variables psf and kel
!
psf = 1.0
kel = 2.0

cnt2d (:)= (/ ilong, ilat /)
strt2d(:)= (/ 1, 1 /)
call Ncwr_2d (psf, ncid, 'psf', strt2d, cnt2d)

PRINT*, ' Successful writing of Surface Pressure'
!
cnt3d (:)= (/ ilong, ilat, invert /)
strt3d(:)= (/ 1, 1, 1 /)
call Ncwr_3d (kel, ncid, 'kel', strt3d, cnt3d)

PRINT*, ' Successful writing of Temperature'
!
! Close the netCDF file
call Nccl(ncid)
!
end

```

2.4 Reading NetCDF Files

Procedures for reading netCDF files can be described as:

```

call Ncop_Rd                      ! Open existing netCDF file
...
call Ncget_Dimlen                  ! Get dimension
...
call Ncrd_2d                      ! Get values of variables
call Nccl                           ! Close netCDF file

```

As example, let us consider the netCDF file produced in the previous subsection. Assume that you want to extract the surface pressure but you do not know the values of "ilong" and "invert".

```
!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!
```

```
! This program extracts the surface pressure (psf, 2D variable)
! available in the file my_filename.nc produced by the program
! creating_my_ncfile.
!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!

      program reading_my_ncfile
!
! Modules from the utility library
use m_netcdf_io_open
use m_netcdf_io_close
use m_netcdf_io_get_dimlen
use m_netcdf_io_read
!
implicit none
include "netcdf.inc"
!
integer :: lon_dim, lat_dim
integer :: cnt2d (2)
integer :: strt2d(2)
integer :: ncid
real*8, allocatable :: psf(:,:)
!
! Open the netCDF file
call Ncop_Rd (ncid, 'my_filename.nc')
!
! Get the dimensions
call Ncget_Dimlen (ncid, 'lon_dim', lon_dim)
call Ncget_Dimlen (ncid, 'lat_dim', lat_dim)

PRINT*, ' Longitude dimension = ', lon_dim
PRINT*, ' Latitude dimension = ', lat_dim
!
! Read the "psf" variable
allocate (psf(lon_dim, lat_dim))
!
strt2d(:) = (/ 1, 1 /)
cnt2d (:)= (/ lon_dim, lat_dim /)
call Ncrd_2d (psf, ncid, 'psf', strt2d, cnt2d)

PRINT*, ' Successful reading of Surface Pressure'
!
! Close netCDF file
call Nccl(ncid)
!
end
```

References

- [1] J. Kouatchou, B. Das and H. Oloso, Global Modeling Initiative: Tutorial and User's Guide, online documentation available at <http://gmi.gsfc.nasa.gov/tutorial/index.html>.
- [2] R. Rew, G. Davis, S. Emmerson and H. Davies, *NetCDF User's Guide for FORTRAN: An Access Interface for Self-Describing, Portable Data*, Unidata Program Center, June 1997.

A Routine/Function Prologues

A.1 Fortran: Module Interface m_Do_Err_Out (Source File: m_do_err_out.f90)

INTERFACE:

```
module m_Do_Err_Out
  implicit none
```

PUBLIC MEMBER FUNCTIONS:

```
public  Do_Err_Out
```

DESCRIPTION:

Provides a routine to print an error message and exit the code.

AUTHOR:

Jules Kouatchou

REVISION HISTORY:

Initial code.

A.1.1 Do_Err_Out

INTERFACE:

```
subroutine Do_Err_Out  &
  (err_msg, err_do_stop, err_num_ints, err_int1, err_int2,  &
   err_num_reals, err_real1, err_real2)
  implicit none
```

INPUT PARAMETERS:

```
!      err_msg        : error message to be printed out
!      err_do_stop    : do stop on error?
!      err_num_ints  : number of integers to be printed out (0, 1, or 2)
!      err_int1       : integer 1 to print out
!      err_int2       : integer 2 to print out
!      err_num_reals : number of reals to be printed out (0, 1, or 2)
!      err_real1     : real 1 to print out
!      err_real2     : real 2 to print out
      character (len=*), intent(in) :: err_msg
      logical                 , intent(in) :: err_do_stop
      integer                , intent(in) :: err_num_ints
      integer                , intent(in) :: err_int1
      integer                , intent(in) :: err_int2
      integer                , intent(in) :: err_num_reals
      real*8                , intent(in) :: err_real1
      real*8                , intent(in) :: err_real2
```

DESCRIPTION:

Outputs error messages, and exits if requested.

AUTHOR:

John Tannahill (LLNL) and Jules Kouatchou

REVISION HISTORY:

Initial code.

A.2 Fortran: Module Interface *m_ncdf_io_checks* (Source File: *m_ncdf_io_checks.f90*)**INTERFACE:**

```
module m_ncdf_io_checks
implicit none
```

PUBLIC MEMBER FUNCTIONS:

```
public Ncdoes_Udim_Exist
public Ncdoes_Var_Exist
```

DESCRIPTION:

Routines to check if a netCDF file contains a specified variable.

AUTHOR:

Jules Kouatchou

REVISION HISTORY:

Initial code.

A.2.1 Ncdoes_Udim_Exist (Source File: *m_ncdf_io_checks.f90*)**INTERFACE:**

```
function Ncdoes_Udim_Exist (ncid)
implicit none
include "netcdf.inc"
```

INPUT PARAMETERS:

```
!    ncid : netCDF file id to check
      integer, intent (in)    :: ncid
```

DESCRIPTION:

Checks a given netCDF file to see if it contains an unlimited dimension.

RETURN VALUE:

```
logical :: Ncdoes_Udim_Exist
```

AUTHOR:

John Tannahill (LLNL) and Jules Kouatchou

REVISION HISTORY:

Initial code.

A.2.2 Ncdoes_Var_Exist (Source File: m_ncdf_io_checks.f90)**INTERFACE:**

```
function Ncdoes_Var_Exist (ncid, varname)
implicit none
include "netcdf.inc"
```

INPUT PARAMETERS:

```
!      ncid      : netCDF file id      to check
!      varname   : netCDF variable name to check
      integer,           intent (in)    :: ncid
      character (len=*), intent (in)    :: varname
```

DESCRIPTION:

Checks a given netCDF file to see if a given netCDF variable exists in it.

RETURN VALUE:

```
logical :: Ncdoes_Var_Exist
```

AUTHOR:

John Tannahill (LLNL) and Jules Kouatchou

REVISION HISTORY:

Initial code.

A.3 Fortran: Module Interface m_ncdf_io_close (Source File: m_ncdf_io_close.f90)

INTERFACE:

```
module m_ncdf_io_close
  implicit none
```

PUBLIC MEMBER FUNCTIONS:

```
public  Nccl
public  Nccl_Noerr
```

DESCRIPTION:

Routines to close a netCDF file.

AUTHOR:

Jules Kouatchou

REVISION HISTORY:

Initial code.

A.3.1 Nccl

INTERFACE:

```
subroutine Nccl (ncid)
```

USES:

```
use m_do_err_out
implicit none
include "netcdf.inc"
```

INPUT PARAMETERS:

```
!      ncid : netCDF file id
      integer, intent (in)    :: ncid
```

DESCRIPTION:

Closes a netCDF file with file id ncid.

AUTHOR:

John Tannahill (LLNL) and Jules Kouatchou

REVISION HISTORY:

Initial code.

A.3.2 Nccl_Noerr

INTERFACE:

```
subroutine Nccl_Noerr (ncid)
implicit none
include "netcdf.inc"
```

INPUT PARAMETERS:

```
!      ncid : netCDF file id
      integer, intent (in)    :: ncid
```

DESCRIPTION:

Closes a netCDF file (with file id ncid) if it is open and suppresses Ncclos error messages/exit if it is not.

AUTHOR:

John Tannahill (LLNL) and Jules Kouatchou

REVISION HISTORY:

Initial code.

A.4 Fortran: Module Interface m_ncdf_io_create (Source File: m_ncdf_io_create.f90)

INTERFACE:

```
module m_ncdf_io_create
implicit none
```

PUBLIC MEMBER FUNCTIONS:

```
public  Nccr_Wr
public  Ncdo_Sync
```

DESCRIPTION:

Routines for creating and syncronizing netCDF files.

AUTHOR:

Jules Kouatchou

REVISION HISTORY:

Initial code.

A.4.1 Nccr_Wr**INTERFACE:**

```
subroutine Nccr_Wr (ncid, filename)
```

USES:

```
use m_do_err_out
implicit none
include "netcdf.inc"
```

INPUT PARAMETERS:

```
!      ncid      : opened netCDF file id
      filename : name of netCDF file to open for writing
      integer            , intent(in)    :: ncid
      character (len=*), intent(in)    :: filename
```

DESCRIPTION:

Creates a netCDF file for writing and does some error checking.

AUTHOR:

John Tannahill (LLNL) and Jules Kouatchou

REVISION HISTORY:

Initial code.

A.4.2 Ncdo_Sync**INTERFACE:**

```
subroutine Ncdo_Sync (ncid)
```

USES:

```
use m_do_err_out
implicit none
include "netcdf.inc"
```

INPUT PARAMETERS:

```
!      ncid : netCDF file id
      integer, intent(in)    :: ncid
```

DESCRIPTION:

Synchronizes a netCDF file.

AUTHOR:

John Tannahill (LLNL) and Jules Kouatchou

REVISION HISTORY:

Initial code.

A.5 Fortran: Module Interface m_ncdf_io_define (Source File: m_ncdf_io_define.f90)

INTERFACE:

```
module m_ncdf_io_define
implicit none
```

PUBLIC MEMBER FUNCTIONS:

```
public NcDef_dimension
public NcDef_variable
public NcDef_var_attributes
public NcDef_glob_attributes
public NcSetFill
public NcEnd_def
```

DESCRIPTION:

Provides netCDF utility routines to define dimensions, variables and attributes.

AUTHOR:

Jules Kouatchou

REVISION HISTORY:

Initial code.

A.5.1 NcDef_dimension

INTERFACE:

```
subroutine NcDef_dimension(ncid,name,len,id)
```

USES:

```
use m_do_err_out
implicit none
include 'netcdf.inc'
```

INPUT PARAMETERS:

```

!      ncid   : netCDF file id
!      name   : dimension name
!      len    : dimension number
      character (len=*), intent(in) :: name
      integer,           intent(in) :: ncid, len

```

OUTPUT PARAMETERS:

```

!      id     : dimension id
      integer,           intent(out) :: id

```

DESCRIPTION:

Defines dimension.

AUTHOR:

Jules Kouatchou and Maharaj Bhat

REVISION HISTORY:

Initial code.

A.5.2 NcDef_variable**INTERFACE:**

```
subroutine NcDef_variable(ncid,name,type,ndims,dims,var_id)
```

USES:

```

use m_do_err_out
implicit none
include 'netcdf.inc'

```

INPUT PARAMETERS:

```

!      ncid   : netCDF file id
!      name   : name of the variable
!      type   : type of the variable
!              (NF_FLOAT, NF_CHAR, NF_INT, NF_DOUBLE, NF_BYTE, NF_SHORT)
!      ndims  : number of dimensions of the variable
!      dims   : netCDF dimension id of the variable
!      varid  : netCDF varid id

      character (len=*), intent(in) :: name
      integer,           intent(in) :: ncid, ndims, var_id
      integer,           intent(in) :: dims(ndims)
      integer,           intent(in) :: type

```

DESCRIPTION:

Defines a netCDF variable.

AUTHOR:

Jules Kouatchou and Maharaj Bhat

REVISION HISTORY:

Initial code.

A.5.3 NcDef_var_attributes**INTERFACE:**

```
subroutine NcDef_var_attributes(ncid,var_id,att_name,att_val)
```

USES:

```
use m_do_err_out
implicit none
include 'netcdf.inc'
```

INPUT PARAMETERS:

```
!      ncid      : netCDF file id
!      var_id    : netCDF variable id
!      att_name: attribute name
!      att_val : attribute value
      character (len=*), intent(in) :: att_name, att_val
      integer,           intent(in) :: ncid, var_id
```

DESCRIPTION:

Defines netCDF attributes.

AUTHOR:

Jules Kouatchou and Maharaj Bhat

REVISION HISTORY:

Initial code.

A.5.4 NcDef_glob_attributes

INTERFACE:

```
subroutine NcDef_glob_attributes(ncid,att_name,att_val)
```

USES:

```
use m_do_err_out
implicit none
include 'netcdf.inc'
```

INPUT PARAMETERS:

```
!      ncid      : netCDF file id
!      att_name: attribute name
!      att_val : attribute value
      character (len=*), intent(in) :: att_name, att_val
      integer,           intent(in) :: ncid
```

DESCRIPTION:

Defines global attributes

AUTHOR:

Jules Kouatchou

REVISION HISTORY:

Initial code.

A.5.5 NcSetFill

INTERFACE:

```
subroutine NcSetFill(ncid,ifill,omode)
```

USES:

```
use m_do_err_out
implicit none
include 'netcdf.inc'
```

INPUT PARAMETERS:

```
integer,           intent(in) :: ncid, ifill,omode
```

DESCRIPTION:

Sets fill method.

AUTHOR:

Jules Kouatchou

REVISION HISTORY:

Initial code.

A.5.6 NcEnd_def

INTERFACE:

```
subroutine NcEnd_def(ncid)
```

USES:

```
use m_do_err_out
implicit none
include 'netcdf.inc'
```

INPUT PARAMETERS:

```
integer,           intent(in) :: ncid
```

DESCRIPTION:

Ends definitions of variables and their attributes.

AUTHOR:

Jules Kouatchou

REVISION HISTORY:

Initial code.

A.6 Fortran: Module Interface *m_ncdf_io_get_dimlen* (Source File: *m_ncdf_io_get_dimlen.f90*)

INTERFACE:

```
module m_ncdf_io_get_dimlen
implicit none
```

PUBLIC MEMBER FUNCTIONS:

```
public  Ncget_Dimlen
public  Ncget_Unlim_Dimlen
```

DESCRIPTION:

Provides routines to obtain the length of a given dimension.

AUTHOR:

Jules Kouatchou

REVISION HISTORY:

Initial code.

A.6.1 Ncget_Dimlen**INTERFACE:**

```
subroutine Ncget_Dimlen (ncid, dim_name, dim_len)
```

USES:

```
use m_do_err_out
implicit none
include 'netcdf.inc'
```

INPUT PARAMETERS:

```
! dim_name : netCDF dimension name
! ncid      : netCDF file id
character (len=*), intent(in) :: dim_name
integer,           intent(in) :: ncid
```

OUTPUT PARAMETERS:

```
! dim_len: netCDF dimension length
integer,          intent(out) :: dim_len
```

DESCRIPTION:

Returns the length of a given netCDF dimension.

AUTHOR:

John Tannahill (LLNL) and Jules Kouatchou

REVISION HISTORY:

Initial code.

A.6.2 Ncget_Unlim_Dimlen**INTERFACE:**

```
subroutine Ncget_Unlim_Dimlen (ncid, udim_len)
```

USES:

```
use m_do_err_out
implicit none
include 'netcdf.inc'
```

INPUT PARAMETERS:

```
! ncid      : netCDF file id
integer,          intent(in) :: ncid
```

OUTPUT PARAMETERS:

```
!     udim_len : netCDF unlimited dimension length
      integer,           intent(out) :: udim_len
```

DESCRIPTION:

Returns the length of the unlimited netCDF dimension.

AUTHOR:

John Tannahill (LLNL) and Jules Kouatchou

REVISION HISTORY:

Initial code.

A.7 Fortran: Module Interface m_ncdf_io_handle_err (Source File: m_ncdf_io_handle_err.f90)**INTERFACE:**

```
module m_ncdf_io_handle_err
  implicit none
```

PUBLIC MEMBER FUNCTIONS:

```
public  Nchandle_Err
```

DESCRIPTION:

Provides a routine to handle error messages.

AUTHOR:

Jules Kouatchou

REVISION HISTORY:

Initial code.

A.7.1 Nchandle_Err**INTERFACE:**

```
subroutine Nchandle_Err (ierr)
```

USES:

```
use m_do_err_out
implicit none
include "netcdf.inc"
```

INPUT PARAMETERS:

```
ierr : netCDF error number
integer, intent (in) :: ierr
```

DESCRIPTION:

Handles netCDF errors. Prints out a message and then exit.

AUTHOR:

John Tannahill (LLNL) and Jules Kouatchou

REVISION HISTORY:

Initial code.

A.8 Fortran: Module Interface m_ncdf_io_open (Source File: m_ncdf_io_open.f90)**INTERFACE:**

```
module m_ncdf_io_open
implicit none
```

PUBLIC MEMBER FUNCTIONS:

```
public Ncop_Rd
public Ncop_Wr
```

DESCRIPTION:

Routines to open a netCDF file.

AUTHOR:

Jules Kouatchou

REVISION HISTORY:

Initial code.

A.8.1 Ncop_Rd**INTERFACE:**

```
subroutine Ncop_Rd (ncid, filename)
```

USES:

```
use m_do_err_out
implicit none
include "netcdf.inc"
```

INPUT PARAMETERS:

```
!     filename : name of netCDF file to open for reading
      character (len=*), intent (in)    :: filename
```

OUTPUT PARAMETERS:

```
!     ncid    : opened netCDF file id
      integer           , intent (out)   :: ncid
```

DESCRIPTION:

Opens a netCDF file for reading and does some error checking.

AUTHOR:

John Tannahill (LLNL) and Jules Kouatchou

REVISION HISTORY:

Initial code.

A.8.2 Ncop_Wr*INTERFACE:*

```
subroutine Ncop_Wr (ncid, filename)
```

USES:

```
use m_do_err_out
implicit none
include "netcdf.inc"
```

INPUT PARAMETERS:

```
!     filename : name of netCDF file to open for reading
      character (len=*), intent (in)    :: filename
```

OUTPUT PARAMETERS:

```
!     ncid    : opened netCDF file id
      integer           , intent (out)   :: ncid
```

DESCRIPTION:

Opens a netCDF file for reading/writing and does some error checking.

AUTHOR:

John Tannahill (LLNL) and Jules Kouatchou

REVISION HISTORY:

Initial code.

A.9 Fortran: Module Interface m_ncdf_io_read (Source File: m_ncdf_io_read.f90)

INTERFACE:

```
module m_ncdf_io_read
  implicit none
```

PUBLIC MEMBER FUNCTIONS:

```
public Ncrd_Scal
public Ncrd_Scal_Int
public Ncrd_1d
public Ncrd_1d_Int
public Ncrd_2d
public Ncrd_2d_Int
public Ncrd_3d
public Ncrd_3d_Int
public Ncrd_4d
public Ncrd_5d
public Ncrd_1d_Char
public Ncrd_2d_Char
```

DESCRIPTION:

Routines for reading variables in a netCDF file.

AUTHOR:

Jules Kouatchou

REVISION HISTORY:

Initial code.

A.9.1 Ncrd_Scal

INTERFACE:

```
subroutine Ncrd_Scal (varrd_scal, ncid, varname)
```

USES:

```
use m_do_err_out
implicit none
include "netcdf.inc"
```

INPUT PARAMETERS:

```
!      ncid       : netCDF file id to read variable from
!      varname    : netCDF variable name
      integer          , intent(in)   :: ncid
      character (len=*) , intent(in)   :: varname
```

OUTPUT PARAMETERS:

```
!     varrd_scal : variable to fill
      real*8           , intent(out)  :: varrd_scal
```

DESCRIPTION:

Reads in a netCDF scalar variable.

AUTHOR:

John Tannahill (LLNL) and Jules Kouatchou

REVISION HISTORY:

Initial code.

A.9.2 Ncrd_Scal_Int*INTERFACE:*

```
subroutine Ncrd_Scal_Int (varrd_scali, ncid, varname)
```

USES:

```
use m_do_err_out
implicit none
include "netcdf.inc"
```

INPUT PARAMETERS:

```
!     ncid       : netCDF file id to read variable from
!     varname    : netCDF variable name
      integer           , intent(in)  :: ncid
      character (len=*), intent(in)  :: varname
```

OUTPUT PARAMETERS:

```
!     varrd_scali : integer variable to fill
      integer           , intent(out)  :: varrd_scali
```

DESCRIPTION:

Reads in a netCDF integer scalar variable.

AUTHOR:

John Tannahill (LLNL) and Jules Kouatchou

REVISION HISTORY:

Initial code.

A.9.3 Ncrd_1d**INTERFACE:**

```
subroutine Ncrd_1d (varrd_1d, ncid, varname, strt1d, cnt1d)
```

USES:

```
use m_do_err_out
implicit none
include "netcdf.inc"
```

INPUT PARAMETERS:

```
!      ncid      : netCDF file id to read array input data from
!      varname   : netCDF variable name for array
!      strt1d    : vector specifying the index in varrd_1d where
!                  the first of the data values will be read
!      cnt1d     : varrd_1d dimension
      integer      , intent(in)    :: ncid
      character (len=*), intent(in)    :: varname
      integer      , intent(in)    :: strt1d(1)
      integer      , intent(in)    :: cnt1d (1)
```

OUTPUT PARAMETERS:

```
!      varrd_1d : array to fill
      real*8       , intent(out)   :: varrd_1d(cnt1d(1))
```

DESCRIPTION:

Reads in a 1D netCDF real array and does some error checking.

AUTHOR:

John Tannahill (LLNL) and Jules Kouatchou

REVISION HISTORY:

Initial code.

A.9.4 Ncrd_1d_Int**INTERFACE:**

```
subroutine Ncrd_1d_Int (varrd_1di, ncid, varname, strt1d, cnt1d)
```

USES:

```
use m_do_err_out
implicit none
include "netcdf.inc"
```

INPUT PARAMETERS:

```
!      ncid      : netCDF file id to read array input data from
!      varname   : netCDF variable name for array
!      strt1d    : vector specifying the index in varrd_1di where
!                  the first of the data values will be read
!      cnt1d     : varrd_1di dimension
integer          , intent(in)  :: ncid
character (len=*), intent(in)  :: varname
integer          , intent(in)  :: strt1d(1)
integer          , intent(in)  :: cnt1d (1)
```

OUTPUT PARAMETERS:

```
!      varrd_1di : integer array to fill
integer          , intent(out) :: varrd_1di(cnt1d(1))
```

DESCRIPTION:

Reads in a 1D netCDF integer array and does some error checking.

AUTHOR:

John Tannahill (LLNL) and Jules Kouatchou

REVISION HISTORY:

Initial code.

A.9.5 Ncrd_2d**INTERFACE:**

```
subroutine Ncrd_2d (varrd_2d, ncid, varname, strt2d, cnt2d)
```

USES:

```
use m_do_err_out
implicit none
include "netcdf.inc"
```

INPUT PARAMETERS:

```
!      ncid      : netCDF file id to read array input data from
!      varname   : netCDF variable name for array
!      strt2d    : vector specifying the index in varrd_2d where
!                  the first of the data values will be read
!      cnt2d     : varrd_2d dimensions
integer          , intent(in)  :: ncid
character (len=*), intent(in)  :: varname
integer          , intent(in)  :: strt2d(2)
integer          , intent(in)  :: cnt2d (2)
```

OUTPUT PARAMETERS:

```
!     varrd_2d : array to fill
      real*8           , intent(out)  :: varrd_2d(cnt2d(1), cnt2d(2))
```

DESCRIPTION:

Reads in a 2D netCDF real array and does some error checking.

AUTHOR:

John Tannahill (LLNL) and Jules Kouatchou

REVISION HISTORY:

Initial code.

A.9.6 Ncrd_2d_Int*INTERFACE:*

```
subroutine Ncrd_2d_Int (varrd_2di, ncid, varname, strt2d, cnt2d)
```

USES:

```
use m_do_err_out
implicit none
include "netcdf.inc"
```

INPUT PARAMETERS:

```
!     ncid      : netCDF file id to read array input data from
!     varname   : netCDF variable name for array
!     strt2d    : vector specifying the index in varrd_2d where
!                  the first of the data values will be read
!     cnt2d     : varrd_2di dimensions
      integer      , intent(in)  :: ncid
      character (len=*), intent(in)  :: varname
      integer      , intent(in)  :: strt2d(2)
      integer      , intent(in)  :: cnt2d (2)
```

OUTPUT PARAMETERS:

```
!     varrd_2di : intger array to fill
      integer      , intent(out)  :: varrd_2di(cnt2d(1), cnt2d(2))
```

DESCRIPTION:

Reads in a 2D netCDF integer array and does some error checking.

AUTHOR:

John Tannahill (LLNL) and Jules Kouatchou

REVISION HISTORY:

Initial code.

A.9.7 Ncrd_3d**INTERFACE:**

```
subroutine Ncrd_3d (varrd_3d, ncid, varname, strt3d, cnt3d)
```

USES:

```
use m_do_err_out
implicit none
include "netcdf.inc"
```

INPUT PARAMETERS:

```
!      ncid      : netCDF file id to read array input data from
!      varname   : netCDF variable name for array
!      strt3d    : vector specifying the index in varrd_3d where
!                  the first of the data values will be read
!      cnt3d     : varrd_3d dimensions
      integer      , intent(in)  :: ncid
      character (len=*) , intent(in)  :: varname
      integer      , intent(in)  :: strt3d(3)
      integer      , intent(in)  :: cnt3d (3)
```

OUTPUT PARAMETERS:

```
!      varrd_3d : array to fill
      real*8       , intent(out)  :: varrd_3d(cnt3d(1), cnt3d(2), &
                                         cnt3d(3))
```

DESCRIPTION:

Reads in a 3D netCDF real array and does some error checking.

AUTHOR:

John Tannahill (LLNL) and Jules Kouatchou

REVISION HISTORY:

Initial code.

A.9.8 Ncrd_3d_Int**INTERFACE:**

```
subroutine Ncrd_3d_Int (varrd_3di, ncid, varname, strt3d, cnt3d)
```

USES:

```
use m_do_err_out
implicit none
include "netcdf.inc"
```

INPUT PARAMETERS:

```
!      ncid      : netCDF file id to read array input data from
!      varname   : netCDF variable name for array
!      strt3d    : vector specifying the index in varrd_3d where
!                  the first of the data values will be read
!      cnt3d     : varrd_3di dimensions
integer           , intent(in)  :: ncid
character (len=*) , intent(in)  :: varname
integer           , intent(in)  :: strt3d(3)
integer           , intent(in)  :: cnt3d (3)
```

OUTPUT PARAMETERS:

```
!      varrd_3di : intger array to fill
integer           , intent(out) :: varrd_3di(cnt3d(1), cnt3d(2), &
                                              cnt3d(3))
```

DESCRIPTION:

Reads in a 3D netCDF integer array and does some error checking.

AUTHOR:

John Tannahill (LLNL) and Jules Kouatchou

REVISION HISTORY:

Initial code.

A.9.9 Ncrd_4d**INTERFACE:**

```
subroutine Ncrd_4d (varrd_4d, ncid, varname, strt4d, cnt4d)
```

USES:

```
use m_do_err_out
implicit none
include "netcdf.inc"
```

INPUT PARAMETERS:

```

!      ncid      : netCDF file id to read array input data from
!      varname   : netCDF variable name for array
!      strt4d    : vector specifying the index in varrd_4d where
!                  the first of the data values will be read
!      cnt4d     : varrd_4d dimensions
      integer      , intent(in)  :: ncid
      character (len=*), intent(in)  :: varname
      integer      , intent(in)  :: strt4d(4)
      integer      , intent(in)  :: cnt4d (4)

```

OUTPUT PARAMETERS:

```

!      varrd_4d : array to fill
      real*8      , intent(out)  :: varrd_4d(cnt4d(1), cnt4d(2), &
                                         cnt4d(3), cnt4d(4))

```

DESCRIPTION:

Reads in a 4D netCDF real array and does some error checking.

AUTHOR:

John Tannahill (LLNL) and Jules Kouatchou

REVISION HISTORY:

Initial code.

A.9.10 Ncrd_5d**INTERFACE:**

```
subroutine Ncrd_5d (varrd_5d, ncid, varname, strt5d, cnt5d)
```

USES:

```

use m_do_err_out
implicit none
include "netcdf.inc"

```

INPUT PARAMETERS:

```

!      ncid      : netCDF file id to read array input data from
!      varname   : netCDF variable name for array
!      strt5d    : vector specifying the index in varrd_5d where
!                  the first of the data values will be read
!      cnt5d     : varrd_5d dimensions
      integer      , intent(in)  :: ncid
      character (len=*), intent(in)  :: varname
      integer      , intent(in)  :: strt5d(5)
      integer      , intent(in)  :: cnt5d (5)

```

OUTPUT PARAMETERS:

```
!     varrd_5d : array to fill
      real*8           , intent(out)  :: varrd_5d(cnt5d(1), cnt5d(2), &
                                         cnt5d(3), cnt5d(4), &
                                         cnt5d(5))
```

DESCRIPTION:

Reads in a 5D netCDF real array and does some error checking.

AUTHOR:

John Tannahill (LLNL) and Jules Kouatchou

REVISION HISTORY:

Initial code.

A.9.11 Ncrd_1d_Char*INTERFACE:*

```
subroutine Ncrd_1d_Char (varrd_1dc, ncid, varname, strt1d, cnt1d)
```

USES:

```
use m_do_err_out
implicit none
include "netcdf.inc"
```

INPUT PARAMETERS:

```
!     ncid      : netCDF file id to read array input data from
!     varname   : netCDF variable name for array
!     strt1d    : vector specifying the index in varrd_1dc where
!                  the first of the data values will be read
!     cnt1d     : varrd_1dc dimension
      integer           , intent(in)  :: ncid
      character (len=*) , intent(in)  :: varname
      integer           , intent(in)  :: strt1d(1)
      integer           , intent(in)  :: cnt1d (1)
```

OUTPUT PARAMETERS:

```
!     varrd_1dc : intger array to fill
      character (len=1), intent(out)  :: varrd_1dc(cnt1d(1))
```

DESCRIPTION:

Reads in a 1D netCDF character array and does some error checking.

AUTHOR:

Jules Kouatchou

REVISION HISTORY:

Initial code.

A.9.12 Ncrd_2d_Char

INTERFACE:

```
subroutine Ncrd_2d_Char (varrd_2dc, ncid, varname, strt2d, cnt2d)
```

USES:

```
use m_do_err_out
implicit none
include "netcdf.inc"
```

INPUT PARAMETERS:

```
!      ncid      : netCDF file id to read array input data from
!      varname   : netCDF variable name for array
!      strt2d    : vector specifying the index in varrd_2dc where
!                  the first of the data values will be read
!      cnt2d     : varrd_2dc dimensions
      integer      , intent(in)   :: ncid
      character (len=*), intent(in)   :: varname
      integer      , intent(in)   :: strt2d(2)
      integer      , intent(in)   :: cnt2d (2)
```

OUTPUT PARAMETERS:

```
!      varrd_2dc : character array to fill
      character      , intent(out)  :: varrd_2dc(cnt2d(1), cnt2d(2))
```

DESCRIPTION:

Reads in a 2D netCDF character array and does some error checking.

AUTHOR:

Jules Kouatchou

REVISION HISTORY:

Initial code.

A.10 Fortran: Module Interface m_ncdf_io_write (Source File: m_ncdf_io_write.f90)

INTERFACE:

```
module m_ncdf_io_write
implicit none
```

PUBLIC MEMBER FUNCTIONS:

```
public Ncwr_Scal
public Ncwr_Scal_Int
public Ncwr_1d
public Ncwr_1d_Int
public Ncwr_2d
public Ncwr_2d_Int
public Ncwr_3d
public Ncwr_3d_Int
public Ncwr_4d
public Ncwr_5d
public Ncwr_6d
public Ncwr_2d_Char
```

DESCRIPTION:

Routines for writing variables in a netCDF file.

AUTHOR:

Jules Kouatchou

REVISION HISTORY:

Initial code.

A.10.1 Ncwr_Scal

INTERFACE:

```
subroutine Ncwr_Scal (varwr_scal, ncid, varname)
```

USES:

```
use m_do_err_out
implicit none
include "netcdf.inc"
```

INPUT PARAMETERS:

```

!      ncid       : netCDF file id to write variable to
!      varname    : netCDF variable name
!      varwr_scal : variable to write out
      integer      , intent(in)   :: ncid
      character (len=*), intent(in)   :: varname
      real*8       , intent(in)   :: varwr_scal

```

DESCRIPTION:

Writes out a netCDF real scalar variable.

AUTHOR:

John Tannahill (LLNL) and Jules Kouatchou

REVISION HISTORY:

Initial code.

A.10.2 Ncwr_Scal_Int**INTERFACE:**

```
subroutine Ncwr_Scal_Int (varwr_scali, ncid, varname)
```

USES:

```

use m_do_err_out
implicit none
include "netcdf.inc"

```

INPUT PARAMETERS:

```

!      ncid       : netCDF file id to write variable to
!      varname    : netCDF variable name
!      varwr_scali : integer variable to write out
      integer      , intent(in)   :: ncid
      character (len=*), intent(in)   :: varname
      integer      , intent(in)   :: varwr_scali

```

DESCRIPTION:

Writes out a netCDF integer scalar variable.

AUTHOR:

John Tannahill (LLNL) and Jules Kouatchou

REVISION HISTORY:

Initial code.

A.10.3 Ncwr_1d**INTERFACE:**

```
subroutine Ncwr_1d (varwr_1d, ncid, varname, strt1d, cnt1d)
```

USES:

```
use m_do_err_out
implicit none
include "netcdf.inc"
```

INPUT PARAMETERS:

```
!      ncid      : netCDF file id to write array output data to
!      varname   : netCDF variable name for array
!      strt1d    : vector specifying the index in varwr_1d where
!                  the first of the data values will be written
!      cnt1d     : varwr_1d dimension
!      varwr_1d : array to write out
      integer      , intent(in)  :: ncid
      character (len=*), intent(in) :: varname
      integer      , intent(in)  :: strt1d(1)
      integer      , intent(in)  :: cnt1d (1)
      real*8       , intent(in)  :: varwr_1d(cnt1d(1))
```

DESCRIPTION:

Writes out a 1D netCDF real array and does some error checking.

AUTHOR:

John Tannahill (LLNL) and Jules Kouatchou

REVISION HISTORY:

Initial code.

A.10.4 Ncwr_1d_Int**INTERFACE:**

```
subroutine Ncwr_1d_Int (varwr_1di, ncid, varname, strt1d, cnt1d)
```

USES:

```
use m_do_err_out
implicit none
include "netcdf.inc"
```

INPUT PARAMETERS:

```

!      ncid      : netCDF file id to write array output data to
!      varname   : netCDF variable name for array
!      strt1d    : vector specifying the index in varwr_1di where
!                  the first of the data values will be written
!      cnt1d     : varwr_1di dimension
!      varwr_1di : integer array to write out
      integer      , intent(in)  :: ncid
      character (len=*) , intent(in)  :: varname
      integer      , intent(in)  :: strt1d(1)
      integer      , intent(in)  :: cnt1d (1)
      integer      , intent(in)  :: varwr_1di(cnt1d(1))

```

DESCRIPTION:

Writes out a 1D netCDF integer array and does some error checking.

AUTHOR:

John Tannahill (LLNL) and Jules Kouatchou

REVISION HISTORY:

Initial code.

A.10.5 Ncwr_2d**INTERFACE:**

```
subroutine Ncwr_2d (varwr_2d, ncid, varname, strt2d, cnt2d)
```

USES:

```

use m_do_err_out
implicit none
include "netcdf.inc"

```

INPUT PARAMETERS:

```

!      ncid      : netCDF file id to write array output data to
!      varname   : netCDF variable name for array
!      strt2d    : vector specifying the index in varwr_2d where
!                  the first of the data values will be written
!      cnt2d     : varwr_2d dimensions
!      varwr_2d : array to write out
      integer      , intent(in)  :: ncid
      character (len=*) , intent(in)  :: varname
      integer      , intent(in)  :: strt2d(2)
      integer      , intent(in)  :: cnt2d (2)
      real*8       , intent(in)  :: varwr_2d(cnt2d(1), cnt2d(2))

```

DESCRIPTION:

Writes out a 2D netCDF real array and does some error checking.

AUTHOR:

John Tannahill (LLNL) and Jules Kouatchou

REVISION HISTORY:

Initial code.

A.10.6 Ncwr_2d_Int**INTERFACE:**

```
subroutine Ncwr_2d_Int (varwr_2di, ncid, varname, strt2d, cnt2d)
```

USES:

```
use m_do_err_out
implicit none
include "netcdf.inc"
```

INPUT PARAMETERS:

```
!      ncid      : netCDF file id to write array output data to
!      varname   : netCDF variable name for array
!      strt2d    : vector specifying the index in varwr_2di where
!                  the first of the data values will be written
!      cnt2d     : varwr_2di dimensions
!      varwr_2di : integer array to write out
      integer      , intent(in)  :: ncid
      character (len=*), intent(in)  :: varname
      integer      , intent(in)  :: strt2d(2)
      integer      , intent(in)  :: cnt2d (2)
      integer      , intent(in)  :: varwr_2di(cnt2d(1), cnt2d(2))
```

DESCRIPTION:

Writes out a 2D netCDF integer array and does some error checking.

AUTHOR:

John Tannahill (LLNL) and Jules Kouatchou

REVISION HISTORY:

Initial code.

A.10.7 Ncwr_3d**INTERFACE:**

```
subroutine Ncwr_3d (varwr_3d, ncid, varname, strt3d, cnt3d)
```

USES:

```
use m_do_err_out
implicit none
include "netcdf.inc"
```

INPUT PARAMETERS:

```
!      ncid      : netCDF file id to write array output data to
!      varname   : netCDF variable name for array
!      strt3d    : vector specifying the index in varwr_3d where
!                  the first of the data values will be written
!      cnt3d     : varwr_3d dimensions
!      varwr_3d : array to write out
      integer      , intent(in)  :: ncid
      character (len=*), intent(in) :: varname
      integer      , intent(in)  :: strt3d(3)
      integer      , intent(in)  :: cnt3d (3)
      real*8       , intent(in)  :: varwr_3d(cnt3d(1), cnt3d(2), cnt3d(3))
```

DESCRIPTION:

Writes out a 3D netCDF real array and does some error checking.

AUTHOR:

John Tannahill (LLNL) and Jules Kouatchou

REVISION HISTORY:

Initial code.

A.10.8 Ncwr_3d_Int**INTERFACE:**

```
subroutine Ncwr_3d_Int (varwr_3di, ncid, varname, strt3d, cnt3d)
```

USES:

```
use m_do_err_out
implicit none
include "netcdf.inc"
```

INPUT PARAMETERS:

```
! ncid      : netCDF file id to write array output data to
! varname   : netCDF variable name for array
! strt3d    : vector specifying the index in varwr_3di where
!             the first of the data values will be written
! cnt3d     : varwr_3di dimensions
! varwr_3di : intger array to write out
integer          , intent(in)  :: ncid
character (len=*), intent(in)  :: varname
integer          , intent(in)  :: strt3d(3)
integer          , intent(in)  :: cnt3d (3)
integer          , intent(in)  :: varwr_3di(cnt3d(1), cnt3d(2), cnt3d(3))
```

DESCRIPTION:

Writes out a 3D netCDF integer array and does some error checking.

AUTHOR:

John Tannahill (LLNL) and Jules Kouatchou

REVISION HISTORY:

Initial code.

A.10.9 Ncwr_4d

INTERFACE:

```
subroutine Ncwr_4d (varwr_4d, ncid, varname, strt4d, cnt4d)
```

USES:

```
use m_do_err_out
implicit none
include "netcdf.inc"
```

INPUT PARAMETERS:

DESCRIPTION:

Writes out a 4D netCDF real array and does some error checking.

AUTHOR:

John Tannahill (LLNL) and Jules Kouatchou

REVISION HISTORY:

Initial code.

A.10.10 Ncwr_5d**INTERFACE:**

```
subroutine Ncwr_5d (varwr_5d, ncid, varname, strt5d, cnt5d)
```

USES:

```
use m_do_err_out
implicit none
include "netcdf.inc"
```

INPUT PARAMETERS:

```
!      ncid      : netCDF file id to write array output data to
!      varname   : netCDF variable name for array
!      strt5d    : vector specifying the index in varwr_5d where
!                  the first of the data values will be written
!      cnt5d     : varwr_5d dimensions
!      varwr_5d : array to write out
      integer          , intent(in)    :: ncid
      character (len=*) , intent(in)    :: varname
      integer          , intent(in)    :: strt5d(5)
      integer          , intent(in)    :: cnt5d (5)
      real*8           , intent(in)    :: varwr_5d(cnt5d(1), cnt5d(2), &
                                         cnt5d(3), cnt5d(4), &
                                         cnt5d(5))
```

DESCRIPTION:

Writes out a 5D netCDF real array and does some error checking.

AUTHOR:

John Tannahill (LLNL) and Jules Kouatchou

REVISION HISTORY:

Initial code.

A.10.11 Ncwr_6d

INTERFACE:

```
subroutine Ncwr_6d (varwr_6d, ncid, varname, strt6d, cnt6d)
```

USES:

```
use m_do_err_out
implicit none
include "netcdf.inc"
```

INPUT PARAMETERS:

```
!      ncid      : netCDF file id to write array output data to
!      varname   : netCDF variable name for array
!      strt6d    : vector specifying the index in varwr_6d where
!                  the first of the data values will be written
!      cnt6d     : varwr_6d dimensions
!      varwr_6d : array to write out
      integer      , intent(in)  :: ncid
      character (len=*), intent(in) :: varname
      integer      , intent(in)  :: strt6d(6)
      integer      , intent(in)  :: cnt6d (6)
      real*8       , intent(in)  :: varwr_6d(cnt6d(1), cnt6d(2), &
                                         cnt6d(3), cnt6d(4), &
                                         cnt6d(5), cnt6d(6))
```

DESCRIPTION:

Writes out a 6D netCDF real array and does some error checking.

AUTHOR:

John Tannahill (LLNL) and Jules Kouatchou

REVISION HISTORY:

Initial code.

A.10.12 Ncwr_1d_Char

INTERFACE:

```
subroutine Ncwr_1d_Char (varwr_1dc, ncid, varname, strt1d, cnt1d)
```

USES:

```
use m_do_err_out
implicit none
include "netcdf.inc"
```

INPUT PARAMETERS:

```

!      ncid      : netCDF file id to write array output data to
!      varname   : netCDF variable name for array
!      strt1d    : vector specifying the index in varwr_1dc where
!                  the first of the data values will be written
!      cnt1d     : varwr_1dc dimension
!      varwr_1dc : integer array to write out
      integer      , intent(in)  :: ncid
      character (len=*) , intent(in)  :: varname
      integer      , intent(in)  :: strt1d(1)
      integer      , intent(in)  :: cnt1d (1)
      character (len=1), intent(in)  :: varwr_1dc(cnt1d(1))

```

DESCRIPTION:

Writes out a 1D netCDF character array and does some error checking.

AUTHOR:

Jules Kouatchou

REVISION HISTORY:

Initial code.

A.10.13 Ncwr_2d_Char*INTERFACE:*

```
subroutine Ncwr_2d_Char (char_2d, ncid, tvarname, strt2d, cnt2d)
```

USES:

```

use m_do_err_out
implicit none
include "netcdf.inc"

```

INPUT PARAMETERS:

```

!      ncid      : netCDF file id to write text to
!      tvarname  : netCDF variable name for text
!      strt2d    : vector specifying the index in char_2d where
!                  the first of the data values will be written
!      cnt2d     : char_2d dimensions
!      char_2d   : text to write
      integer      , intent(in)  :: ncid
      character (len=*) , intent(in)  :: tvarname
      integer      , intent(in)  :: strt2d(2)
      integer      , intent(in)  :: cnt2d (2)
      character (len=1), intent(in)  :: char_2d(cnt2d(1), cnt2d(2))

```

DESCRIPTION:

Writes out a 2D netCDF character array and does some error checking.

AUTHOR:

John Tannahill (LLNL) and Jules Kouatchou

REVISION HISTORY:

Initial code.